Time-Optimal Tube Following for Robotic Manipulators*

Frederik Debrouwere*, Wannes Van Loock*, Goele Pipeleers* and Jan Swevers*
*Department of Mechanical Engineering, Division PMA, KU Leuven, BE-3001 Heverlee, Belgium,
<firstname>.<lastname>@mech.kuleuven.be

***Abstract*—Time-optimal *path following* for robots considers the problem of moving along a predetermined Cartesian geometric path in minimum time. In practice this path need not be followed exactly, but within a certain tolerance; so that the motion may be executed faster. In this paper, we define this deviation as a tube around the given geometric path. This transforms the path following problem into a *tube following* problem. However, unlike the former, the latter is not convex.We propose a problem formulation that can still be solved efficiently, as illustrated by some numerical examples.**

I. Introduction

Robot path following problems determine the trajectory of a robot along a predetermined geometric path without any preassigned timing information. This is often considered to be the low level stage in a decoupled motion planning approach [1]–[3], since the motion planning problem (path and trajectory planning) is difficult and highly complex to solve in its entirety [4], [5]. First, a high level path planner determines a geometric path, ignoring the system dynamics but taking into account geometric path constraints. Second, an optimal trajectory along the geometric path is determined that takes the system dynamics and limitations into account. Since the dynamics along a geometric path can be described by a scalar path coordinate s and its time derivatives [1]–[3], the decoupled approach simplifies the motion planning problem to great extent. Furthermore, the path following problem for a robotic manipulator with simplified constraints can be cast as a *convex* optimization problem [6], [7]. This guarantees efficient computation of globally optimal solutions.

In many applications, the Cartesian geometric end effector path planned by the path planner does not need to be followed exactly but within certain position and orientation tolerances. By deviating from the predetermined path, within the allowable tolerance one could gain in time-optimality. Typical examples are milling robots where some geometrical tolerance on the workpiece is allowed. In this paper we define a tube around the given geometric path, representing the allowable deviation from this path, while we impose constraints on the end effector position, such that it lies inside this tube. We do not consider orientation constraints on the end effector although including them is straightforward. To allow the end effector path to be chosen freely inside this tube we need to allow freedom on the joint paths within some bounds. Furthermore, we need to allow a lot of freedom, much more than needed, to make sure that the tube constraints are the restricting factor in the optimization, and not the joint path bounds. The *tube following* problem we end up with is nonconvex. In this paper we propose a problem formulation, starting from the path following formulation, that can still be solved efficiently using a standard interior point solver.

This paper is organised as follows. Section II reviews the path following problem formulation. Then, Section III extends this path following problem to a tube following problem. Here we define a parametrisation for the path and we define tube constraints on the position of the end effector. Section IV illustrates the proposed framework with some numerical examples.

Throughout the paper we will use the following shorthand notations for the derivatives of a function $f(s(t))$: $\dot{f} = \frac{df}{dt}, \ddot{f} = \frac{d^2f}{dt^2}, f' = \frac{\partial f}{\partial s}, f'' = \frac{\partial^2 s}{\partial s^2}$ where t indicates time and s the path coordinate. Furthermore, we indicate scalars with a lower-case letter, e.g. n, vectors with a bold lower-case letter, e.g. $\boldsymbol{q}$, and matrices with an upper-case letter, e.g. M. $\boldsymbol{q}_i$ denotes the i-th element of $\boldsymbol{q}$. We indicate a surface in 3D Euclidean space with a bold calligraphic letter, e.g. $\boldsymbol{\mathcal{L}}$.

II. Path Following Problem Formulation

Consider a robotic manipulator with n degrees of freedom and joint angles $\boldsymbol{q} \in \mathbb{R}^n$. The equations of motion are given by

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \qquad (1)$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ are the joint torques, $M \in \mathbb{R}^{n\times n}$ is the mass matrix, $C \in \mathbb{R}^{n\times n}$ is a matrix, linear in $\dot{\boldsymbol{q}}$, accounting for Coriolis and centrifugal effects and, $\boldsymbol{g}$ is a vector accounting for gravity and other position dependent torques.

Consider a prescribed geometric path $\boldsymbol{q}(s)$ as a function of a scalar path coordinate s, given in joint space coordinates. The time dependence of the path is determined through $s(t)$. Without loss of generality it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and, $0 = s(0) \leq s(t) \leq s(T) = 1$. It is furthermore assumed that we always move forward along the path, i.e. $\dot{s}(t) \geq 0, \forall t \in [0, T]$.

Using the chain-rule we rewrite joint velocities and accelerations as

$$\dot{\boldsymbol{q}}(s) = \boldsymbol{q}'(s)\dot{s} \text{ and, } \ddot{\boldsymbol{q}}(s) = \boldsymbol{q}''(s)\dot{s}^2 + \boldsymbol{q}'(s)\ddot{s}. \qquad (2)$$

*This work benefits from K.U.Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO), the European research project EMBOCON FP7-ICT-2009-4 248940, project G.0377.09 of the Research Foundation – Flanders (FWO – Vlaanderen), the European Project TEMPO Marie Curie Initial Training Network (607957), and K.U.Leuven's Concerted Research Action GOA/10/11. Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen).

Substitution of the above equations in (1) projects the equations of motion onto the path [6]:

$$\boldsymbol{\tau}(s(t)) = \boldsymbol{\psi}_s \left(s(t), \dot{s}(t)^2, \ddot{s}(t), \boldsymbol{q}(s(t)), \boldsymbol{q}'(s(t)), \boldsymbol{q}''(s(t))\right). \tag{3}$$

Now, by using the same transformation of variables as in [3], [6] we transform the problem from a time t dependent problem into a solely path s dependent problem where we use s as an independent variable instead of time t.

$$\dot{s}^2 = b(s), \text{ where } \ddot{s} = \frac{1}{2}b'(s). \tag{4}$$

This results in the following dynamics

$$\boldsymbol{\tau}(s) = \boldsymbol{\psi}_b \left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right). \tag{5}$$

The total motion time is given by

$$T = \int_0^T 1dt = \int_0^1 \frac{1}{\dot{s}} ds = \int_0^1 \frac{1}{\sqrt{b(s)}} ds,$$

The time-optimal path following problem is then formulated as

$$\begin{aligned} \underset{b(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\ & \boldsymbol{\tau}(s) = \boldsymbol{\psi}_b \left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right) \\ & \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\ & \text{for } s \in [0, 1]. \end{aligned} \tag{6}$$

Once the optimal solution for $b(\cdot), \boldsymbol{\tau}(\cdot)$ is obtained, the relation between path coordinate and time can be obtained from the relation

$$t(s) = \int_0^s \frac{1}{\sqrt{b(\sigma)}} d\sigma.$$

Note that optimization problem (6) is a fixed end-time problem due to the transformation from time domain t to path domain s. In general this is much easier to solve than a free end-time problem due to the strongly non-linear dependence of the solution with varying end-times.

For path following problems, and hence for predetermined geometric paths $\boldsymbol{q}(s)$, the optimization problem is *convex* for a simplified robot and simple task constraints [6], [8]. In the extension to tube following we will allow deviations from the fixed path, hence $\boldsymbol{q}(s)$ is free and the optimization problem is *nonconvex*. The proposed problem formulation, given in the following section, results in an numerical optimization problem which can be solved efficiently using standard nonconvex solvers.

III. From Path Following to Tube Following

Generally a robot task is specified in Cartesian coordinates of the end effector pose $\boldsymbol{y}(s) = (x, y, z, \phi, \theta, \psi)^T = (\boldsymbol{x}(s)^T, \boldsymbol{\phi}(s)^T)^T$, since this is the practical execution space of the robot, while the joint space $\boldsymbol{q}(s)$ coordinates are used to control the robot. Here $\boldsymbol{x} = (x, y, z)^T$ represents the Cartesian position of the end effector in the world frame, and $\boldsymbol{\phi} = (\phi, \theta, \psi)^T$ represents its roll, pitch and yaw angles [9]. The relation between the end effector coordinates and joint coordinates is given by the *forward kinematics*

$$\boldsymbol{y}(s) = \boldsymbol{\chi}(\boldsymbol{q}(s)). \tag{7}$$

We can also define forward position and orientation kinematics as

$$\boldsymbol{x}(s) = \boldsymbol{\chi}_x(\boldsymbol{q}(s)) \text{ and } \boldsymbol{\phi}(s) = \boldsymbol{\chi}_\phi(\boldsymbol{q}(s)).$$

In a path following task a fixed geometric path for $\boldsymbol{q}(s)$ is determined from the desired Cartesian path $\boldsymbol{y}(s)$ by using the inverse relation of $\boldsymbol{\chi}(\cdot)$ (inverse kinematics). In a tube following task, the Cartesian path should lie inside a tube around some predetermined Cartesian path while changes in the orientation from the desired orientation are constrained as well. Hence, we search for a new $\boldsymbol{q}(s)$ such that $\boldsymbol{y}(s)$ lies inside that tube and obeys the orientation constraints. Hence $\boldsymbol{q}(s)$ is a free optimization variable that has to obey constraints on $\boldsymbol{x}(s)$ and $\boldsymbol{\phi}(s)$: $\boldsymbol{x}(s) = \boldsymbol{\chi}_x(\boldsymbol{q}(s)) \in \mathcal{C}_x$ and $\boldsymbol{\phi}(s) = \boldsymbol{\chi}_\phi(\boldsymbol{q}(s)) \in \mathcal{C}_\phi$ for $s \in [0, 1]$, and hence constraints on $\boldsymbol{y}(s) = (\boldsymbol{x}(s)^T, \boldsymbol{\phi}(s)^T)^T$:

$$\boldsymbol{y}(s) = \boldsymbol{\chi}(\boldsymbol{q}(s)) \in \mathcal{C} = \mathcal{C}_x \times \mathcal{C}_\phi \text{ for } s \in [0, 1], \tag{8}$$

In this paper we only consider constraints on the position, hence $\mathcal{C}_x$, and do not consider constraints on the orientation, hence $\mathcal{C}_\phi$. Since $\boldsymbol{\chi}(\cdot)$ is an analytical relation between joint space and Cartesian space it can easily be included in the optimization problem:

$$\begin{aligned} \underset{b(\cdot),\boldsymbol{q}(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\ & \boldsymbol{\tau}(s) = \boldsymbol{\psi}_b \left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right) \\ & \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\ & \boldsymbol{\chi}(\boldsymbol{q}(s)) \in \mathcal{C} \\ & \text{for } s \in [0, 1]. \end{aligned} \tag{9}$$

Since the resulting time-optimal path following problem has an infinite number of optimization variables and an infinite number of constraints, it is discretized by adopting the direct transcription method from [6] which presents a parametrisation for $b(s)$ and $\boldsymbol{\tau}(s)$. Here a discretized s-grid is assumed where we have s^k for $k = 0..K$ and $s^{k+1} = s^k + \Delta s^k$ and $s^{k+\frac{1}{2}} = s^k + \frac{1}{2}\Delta s^k$. Now we need to determine a parametrisation for $\boldsymbol{q}(s)$.

In the following Subsections we review an efficient parametrisation for $\boldsymbol{q}(s)$, already proposed in [10] and present efficient tube constraint formulations determining the set $\mathcal{C}$.

A. Parametrisation of $\boldsymbol{q}(s)$

The path $\boldsymbol{q}_i(s)$ of joint i is defined as a convex combination of predetermined bounds $\underline{\boldsymbol{q}}_i$ and $\overline{\boldsymbol{q}}_i$ for $\boldsymbol{q}_i$ as in [10]

$$\boldsymbol{q}_i(s) = \boldsymbol{p}_i(s)\underline{\boldsymbol{q}}_i(s) + (1 - \boldsymbol{p}_i(s))\overline{\boldsymbol{q}}_i, \tag{10}$$

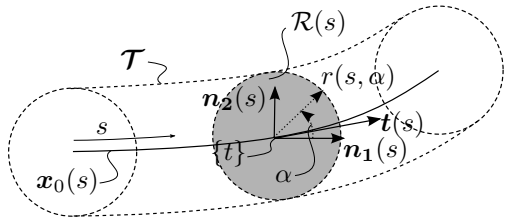

Fig. 1. Tube $\boldsymbol{\mathcal{T}}$ around a central path $\boldsymbol{x}_0(s)$

where $0 \leq \boldsymbol{p}_i(s) \leq 1 \quad \forall s \in [0,1]$ and $\boldsymbol{q}'(s) = \boldsymbol{\alpha}(s, \boldsymbol{p}(s), \boldsymbol{p}'(s))$ and $\boldsymbol{q}''(s) = \boldsymbol{\beta}(s, \boldsymbol{p}(s), \boldsymbol{p}'(s), \boldsymbol{p}''(s))$ can be computed using the chain rule. By choosing the paths $\underline{\boldsymbol{q}}_i$ and $\overline{\boldsymbol{q}}_i$ the torque and end effector pose only depend on $\boldsymbol{p}(s)$, allowing us to write the tube following problem as:

$$\begin{aligned}
\underset{b(\cdot),\boldsymbol{p}(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\
\text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\
& \boldsymbol{\tau}(s) = \boldsymbol{\psi}_p\left(s, b(s), b'(s), \boldsymbol{p}(s), \boldsymbol{p}'(s), \boldsymbol{p}''(s)\right) \\
& 0 \leq \boldsymbol{p}_i(s) \leq 1 \\
& \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\
& \boldsymbol{\chi}(\boldsymbol{q}(\boldsymbol{p}(s))) \in \mathcal{C} \\
& \text{for } s \in [0,1].
\end{aligned}$$

The optimization now does not need to determine a feasible $\boldsymbol{q}(s)$ itself but only needs to determine a convex combination of already predefined bounds $\underline{\boldsymbol{q}}_i$ and $\overline{\boldsymbol{q}}_i$, which are chosen such that they do not become restricting over the tube constraint. The choice of $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ and the parametrisation for $\boldsymbol{p}(s)$ are now design parameters. As shown in Section IV, choosing e.g. $\boldsymbol{p}_i(s)$ to be constant over the whole s range already gives some freedom in the optimization to gain in time-optimality with minimal extra computational cost of the optimization.

B. Tube (Position) Constraints

The actual end effector path $\boldsymbol{x}(s) = (x(s), y(s), z(s))^T$ should lie within a tube around the central Cartesian path $\boldsymbol{x}_0(s) = (x_0(s), y_0(s), z_0(s))^T$ with radius $r(s)$ which represents the allowable deviation from the path. This tube surface $\boldsymbol{\mathcal{T}}$ is described by the following parametric equation:

$$\boldsymbol{\mathcal{T}}(s,\alpha) = \boldsymbol{x}_0(s) + r(s,\alpha)\left(cos(\alpha)\boldsymbol{n_1}(s) + sin(\alpha)\boldsymbol{n_2}(s)\right), \quad \text{for } \alpha \in [0, 2\pi] \text{ and } s = [0,1], \quad (11)$$

with $\boldsymbol{n_1}(s) = (\boldsymbol{t}(s) \times \boldsymbol{v}) / \left\|\boldsymbol{t}(s) \times \boldsymbol{v}\right\|_2$ the normal vector and $\boldsymbol{n_2}(s) = (\boldsymbol{t}(s) \times \boldsymbol{n_1}(s)) / \left\|\boldsymbol{t}(s) \times \boldsymbol{n_1}(s)\right\|_2$ the binormal vector to the tangent $\boldsymbol{t}(s)$ of the central path $\boldsymbol{x}_0(s)$ at s (where $\boldsymbol{v}$ is an arbitrary vector that determines the direction of $\boldsymbol{n_1}$). α represents the angle around the tangent $\boldsymbol{t}$ of the curve $\boldsymbol{x}_0(s)$ at s, starting from the normal vector $\boldsymbol{n_1}$. $r(s,\alpha)$ represents the radius of the tube as a function of s and α. The dependence of $r(s,\alpha)$ on α can be used to allow more freedom in some directions than in others (as will be illustrated in Section IV-B). In the following of this Section we assume r to be independent from α, hence, we assume a constant radius $r(s)$ at every s. Figure 1 clarifies the above definitions with an illustration.

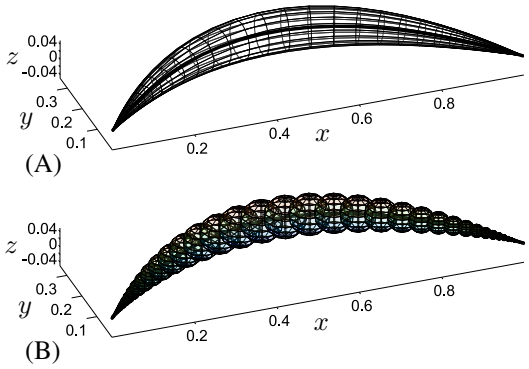

Fig. 2. Illustration of tube approximations: linear (A) and quadratic (B).

The constraint that $\boldsymbol{x}(s)$ should lie in the tube (11) can now be written as:

$$\boldsymbol{x}(s) = \boldsymbol{\chi}_x(\boldsymbol{q}(s)) \quad (12a)$$

$$\boldsymbol{x}(s) = \boldsymbol{x}_0(s) + \hat{r}(s)\left(cos(\hat{\alpha})\boldsymbol{n_1}(s) + sin(\hat{\alpha})\boldsymbol{n_2}(s)\right) \quad (12b)$$

$$\hat{r}(s) \leq r(s) \quad (12c)$$

$$\forall s \in [0,1].$$

by introducing the new optimization variables $\hat{r}(s)$ and $\hat{\alpha}(s)$ while $\hat{\alpha} \in 0, \ldots, 2\pi$. This tube constraint requests that for every s, the point $\boldsymbol{x}(s)$ lies in a disc $\mathcal{R}(s)$ (defined by $r(s^k, \alpha)$ for $\alpha = 0, \ldots, 2\pi$), in the plane orthogonal to the curve $x_0(s^k)$ in s^k. This is illustrated in Figure 1.

The constraints (12) are not very practical to work with in practice. Implementing the tube constraints (12) at each s^k on a discretized s-grid imposes that $\boldsymbol{x}(s^k)$ should lie in the disc $\mathcal{R}(s^k)$ (see Figure 1), which would be to restrictive, since only a limited amount of parametrisations $\boldsymbol{q}(s)$ validate these constraints. This can be seen in Figure 3. This Figure shows the points $\boldsymbol{x}(\boldsymbol{q}(s^k))$, as dots, for a two dof planar robot (given in Section IV-A), where $\boldsymbol{q}$ is defined according to (10) where $\boldsymbol{p}_i = p$ for all i, for a grid of p-values $\in [0,1]$. Here the grey shades of the dots indicate the value of p, where white equals $p = 1.0$ and black $p = 0.0$. It can be seen that only a single p-value ($p = 0.5$, hence $\boldsymbol{x} = \boldsymbol{x}_0$) renders feasible constraints (12). To omit this problem we extend the disc to a volume that approximates the interior of the tube around the curve $\boldsymbol{x}_0(s^k)$ in s^k.

To achieve this we propose two types of approximation approaches: a linear and quadratic approximation, resulting in linear or quadratic constraints on $\boldsymbol{x}(s)$ respectively. Before we continue, we first define a plane $\boldsymbol{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ as the plane parallel to the vectors $\boldsymbol{u}, \boldsymbol{v}$ and going through the point $\boldsymbol{w}$.

1) Linear tube approximation and constraints: The linear approximation method approximates the interior of the tube $\boldsymbol{\mathcal{T}}(s^k, \alpha)$ at s^k by the interior of the polyhedron $\boldsymbol{\mathcal{P}}^k$: $int(\boldsymbol{\mathcal{T}}(s^k, \alpha)) \approx int(\boldsymbol{\mathcal{P}}^k)$.

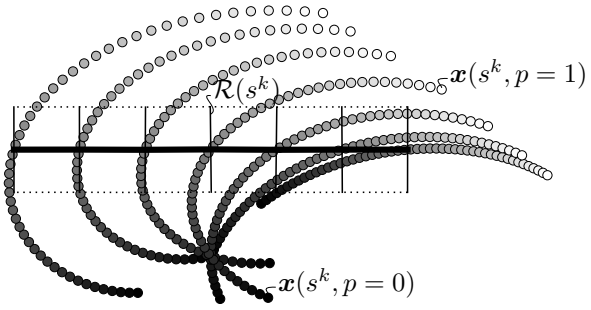

Fig. 3. 2D tube (dotted lines) around a path $\boldsymbol{x}_0(s)$ (thick full line) for a planar robot, illustrating the location of the discs $\mathcal{R}(s^k)$ (thin full lines) and the points $\boldsymbol{x}(s^k, p)$ (dots) for a grid of $p \in [0, 1]$.

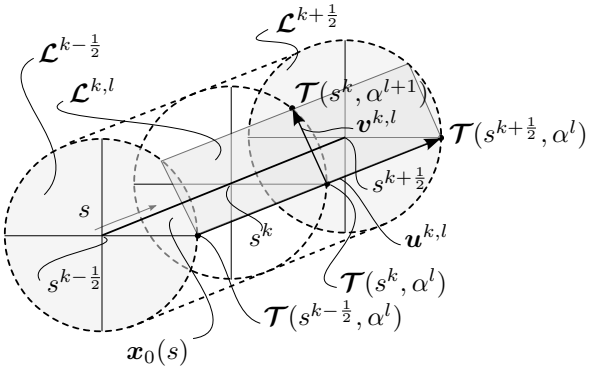

Fig. 4. Linear tube approximation for a section of the tube around s^k in between $s^{k-\frac{1}{2}}$ and $s^{k+\frac{1}{2}}$.

A possible choice for the polyhedron $\mathcal{P}^k$ is a composition of a linear approximation of the tube surface $\mathcal{T}(s^k, \alpha)$ by L planes $\mathcal{L}^{k,l}\left(\boldsymbol{u}^{k,l}, \boldsymbol{v}^{k,l}, \mathcal{T}(s^{k-\frac{1}{2}}, \alpha^l)\right)$ for $l = 0..L-1$ and the planes $\mathcal{L}^{k-\frac{1}{2}}\left(\boldsymbol{n_1}(s^{k-\frac{1}{2}}), \boldsymbol{n_2}(s^{k-\frac{1}{2}}), \boldsymbol{x}_0(s^{k-\frac{1}{2}})\right)$ and $\mathcal{L}^{k+\frac{1}{2}}\left(\boldsymbol{n_1}(s^{k+\frac{1}{2}}), \boldsymbol{n_2}(s^{k+\frac{1}{2}}), \boldsymbol{x}_0(s^{k+\frac{1}{2}})\right)$, cutting the tube in sections at $s^{k-\frac{1}{2}}$ and at $s^{k+\frac{1}{2}}$. Here $\boldsymbol{u}^{k,l} = \mathcal{T}(s^{k+\frac{1}{2}}, \alpha^l) - \mathcal{T}(s^{k-\frac{1}{2}}, \alpha^l)$, $\boldsymbol{v}^{k,l} = \mathcal{T}(s^k, \alpha^{l+1}) - \mathcal{T}(s^k, \alpha^l)$ and $\alpha^l = \frac{2\pi l}{L-1}$ (for $l = 0..L-1$) are discrete values for α. The above definitions are clarified in Figure 4 with an illustration for s^k. The linear tube surface approximation can be seen in Figure 2.A.

The Cartesian point $\boldsymbol{x}(s^k)$ then should lie in the interior of the polytope $\mathcal{P}^k$. This can be written as linear constraints of the form (8) where

$$\mathcal{C}_x = \{\boldsymbol{x}(s^k) | C^k(s^k)\boldsymbol{x}(s^k) + \boldsymbol{d}^k(s^k) \leq 0\},$$

where the matrix $C^k \in \mathbb{R}^{L+2,3}$ and vector $\boldsymbol{d}^k \in \mathbb{R}^{L+2,1}$ are constructed of the plane equations that describe the polytope $\mathcal{P}^k$.

a) approximation error: As can be seen from Figure 5.A, the linear approximation either overestimates or underestimates the tube surface. This approximation error depends on the curvature of $r(s^k)$ and the grid size Δs^k. The smaller the grid size, the smaller the error. Hence, this results in obvious choices for the grid size, discussed in the following paragraph.

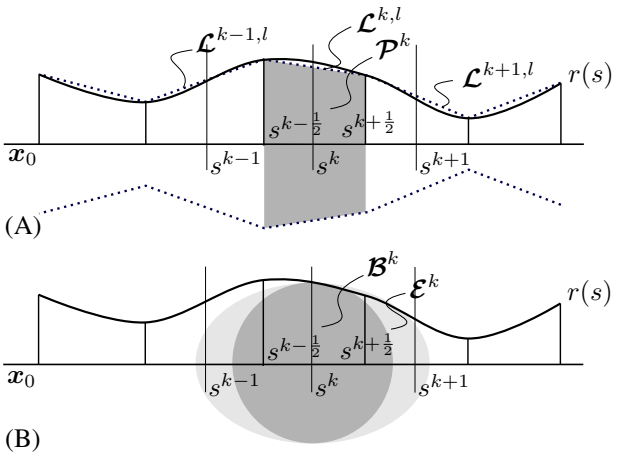

Fig. 5. Tangential sectional view of the tube for the, (A) linear approximation method and (B) quadratic approximation method.

b) choice of the gridsize: The higher the curvature of $r(s)$, the closer the gridpoints s^k should be placed together to obtain a better linear approximation of the tube surface. Hence we get

$$\Delta s^k \sim \left(\frac{d^2 r(s)}{ds^2}\bigg|_{s^k}\right)^{-1}.$$

2) Quadratic tube approximation and constraints: We approximate the tube $\mathcal{T}(s^k, \alpha)$ at every s^k as a sphere $\boldsymbol{\mathcal{B}}^k$ with radius $r(s^k)$. This can be seen in Figure 2.B and 5.B. Hence the request that $\boldsymbol{x}(s^k)$ should lie inside the tube for all s^k can be approximated to $\boldsymbol{x}(s^k)$ lying in $\boldsymbol{\mathcal{B}}^k$. This results in the following quadratic constraint of the form (8) where

$$\mathcal{C}_x = \{\boldsymbol{x}(s^k) | \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right)^T \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right) \leq r(s^k)^2\}.$$

a) extension to ellipsoidal tube approximation: We approximate the tube at every s^k as an ellipsoid. This results in a constraint of the form (8) where

$$\mathcal{C}_x = \{\boldsymbol{x}(s^k) | \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right)^T E^k(s^k) \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right) \leq 1\},$$

where $E^k(s^k)$ is a symmetric positive definite matrix. The eigenvectors of E^k define the principal directions of the ellipsoid while the inverse of the square root of the eigenvalues of E^k are the corresponding equatorial radii. Following the tube definition of (11) the eigenvectors of E^k should be the tangent vector $\boldsymbol{t}$ and normal vectors $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ to the central path $\boldsymbol{x}_0(s^k)$ with radii $r_{\boldsymbol{t}}$, $r_{\boldsymbol{n_1}}$ and $r_{\boldsymbol{n_2}}$. Here $r_{\boldsymbol{n_1}}$ and $r_{\boldsymbol{n_2}}$ define an ellipsoidal cross section $r(s^k, \alpha)$ while $r_{\boldsymbol{t}}$ represents the radius along $\boldsymbol{t}$.

b) approximation error: It can be seen from Figure 5.B that when there is high curvature of $r(s^k)$, the sphere $\boldsymbol{\mathcal{B}}^k$ will not represent the tube surface well at e.g. $s^{k-\frac{1}{2}}$. This high approximation error could be reduced by placing the grid

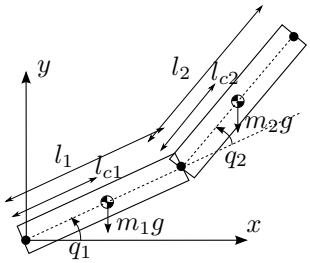

Fig. 6. The planar robot used in the experiments.

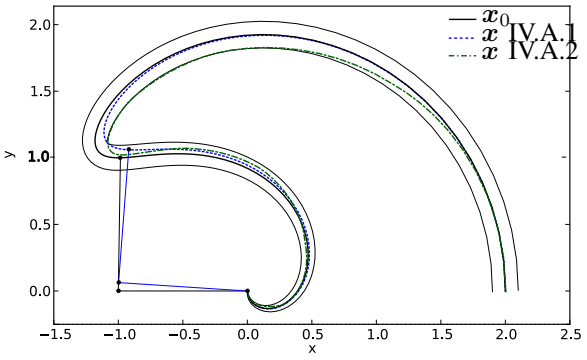

Fig. 7. Results of numerical simulations. Path following path (black) and tube following path (dashed), for two cases, within the tube (black thin lines).

points closely together (see the next paragraph) or by using an ellipsoidal approximation $\boldsymbol{\mathcal{E}}^k$ instead of a spherical. Then the tangential radius $r_{\boldsymbol{t}}$ can be chosen to be bigger when there is small curvature or smaller when there is larger curvature at s^k, to obtain a better approximation of the tube surface (see Figure 5.B). This results in the following guideline:

$$r_{\boldsymbol{t}} \sim \left(\frac{d^2 r(s)}{ds^2}\bigg|_{s^k} \right)^{-1}.$$

c) choice of the gridsize: Since the tube at s^{k-1} and s^k is approximated by spheres with radius $r(s^{k-1})$ and $r(s^k)$, the grid points should be placed closely if the radius $r(s)$ is small to obtain sufficient overlapping of both spheres, to obtain a sufficient approximation of the interior of the tube. If the radius is larger, the grid points can be placed farther away. Furthermore, similarly to the linear case, the gridpoints should be placed closely together when there is high curvature of $r(s)$. Hence we get

$$\Delta s^k \sim r(s^k), \left(\frac{d^2 r(s)}{ds^2}\bigg|_{s^k} \right)^{-1}.$$

Figure 2.B shows the quadratic approximation for a fine s-grid, it can be seen that for a coarser s-grid, the overlapping of the spheres can be insufficient to represent the interior of the tube. Hence, we could request

$$\Delta s^k < \frac{1}{2}\left(r(s^k) + r(s^{k+1})\right).$$

3) Linear vs Quadratic: Both the linear and quadratic approximation have advantages and disadvantages. The linear approximation results in a large amount of constraints ($\geq L$) for each s^k, where the quadratic approximation only has one constraint for each s^k. The quadratic approximation can however only handle circular and elliptical tube cross sections while the linear can handle much complexer cross sections. With respect to approximation error and freedom in the optimization the linear approximation is preferable over the quadratic. Furthermore, a comparison in computation time and convergence of the optimization algorithm should be made. A choice between both approximation methods should be made based on the considered application.

IV. Numerical examples

To illustrate the efficiency of our approach, we consider some examples.

A. planar 2D robot

For simplicity and clarity of the graphs, we consider a 2 dof planar robotic manipulator, which can be seen in Figure 6. The system parameters we consider are the following: $m_1 = 1$ kg, $m_2 = 1$ kg, $l_1 = 1$ m, $l_2 = 1$ m, $l_{c1} = 0.5$ m, $l_{c2} = 0.5$ m, $\boldsymbol{\tau}_- = -(30, 10)^T$ Nm, $\boldsymbol{\tau}_+ = (30, 10)^T$ Nm . We consider the following central path

$$\boldsymbol{q}_0(s) = \begin{pmatrix} -4\pi(s^2 - s), & -\pi s \end{pmatrix}^T,$$

with tube radius $r(s) = 0.1(1 - s^4)$ m. This tube is approximation by the quadratic approximation method where we consider circles (since we work in 2D) and not ellipsoids for simplicity of the example. The yaw angle of the end effector is given by $\psi = \boldsymbol{q}_1 + \boldsymbol{q}_2$. Furthermore, we choose two bounds $\underline{\boldsymbol{q}}(s) = 0.5\boldsymbol{q}_0(s)$ and $\overline{\boldsymbol{q}}(s) = 2\boldsymbol{q}_0(s)$.

The time-optimal tube following problem (9) is implemented in Python using CasADi [11] as modelling software and Ipopt [12] as non-linear solver. All problems are discretized with $K = 100$ and are solved on an Ubuntu Virtual Box with 1Gb RAM, running on a 2.4GHz Windows laptop.

1) Only tube constraints, constant $\boldsymbol{p}(s)$: We consider the planar robot with given tube radius. As a convex path combination parameter $\boldsymbol{p}(s)$ we choose a constant p_i for each joint. The initialisation of $\boldsymbol{p}$ is such that (10) is equal to $\boldsymbol{q}_0(s)$. Figure 7 shows the result of the optimization. The thick, full black path represents the central path $\boldsymbol{x}_0(s)$, the thin full black lines represent the tube boundary, the thick blue dashed line is the optimized path $\boldsymbol{x}(s)$. The two thin lines, connected by dots, represent the robot. We can see that the optimized path hits the tube boundary at only one s-coordinate. This is due to the choice for only a single constant p_i over the whole s-range. Compared to the motion time of the path following case the tube following motion is 1.3% faster. The computation time is however 7 times larger ($1.072s \leftrightarrow 0.152s$), which is still sufficiently fast for a motion planning problem.

2) Only tube constraints, piecewise linear $\boldsymbol{p}(s)$: We consider the same problem but with a piecewise linear $\boldsymbol{p}_i(s)$ for each joint with 10 intervals. This will allow for more freedom in the optimization. Figure 7 shows the result of the optimization (thick green dashed line). We can see that the optimized path

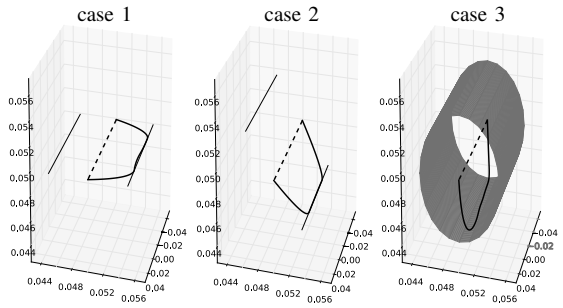

Fig. 8. Results of numerical simulations. Path following path (dashed line) and tube following path (full thick line), for three cases, within the tube (full thin lines).

now hits the tube boundary at much more points due to the piecewise linear $\boldsymbol{p}(s)$. Compared to the motion time of the path following case the tube following motion is 6.2% faster, which is faster than in the first example, due to the piecewise linear $\boldsymbol{p}$. The computation time is 14 times larger ($2.176 \leftrightarrow 0.152s$).

B. Three dof PHANToM(TM) Robot

We consider the three dof PHANToM(TM)[1] Robot of which the kinematics and dynamics are given in [13]. This robot allows 3D Cartesian positioning of the end effector. We consider a Cartesian line (from (-0.05,0.05,0.05) to (0.05,0.05,0.05)) as the central path $\boldsymbol{x}_0(s)$, using the inverse kinematics a relation for $\boldsymbol{q}_0(s)$ can be obtained. We choose two bounds $\underline{\boldsymbol{q}}(s) = 0.5\boldsymbol{q}_0(s)$ and $\overline{\boldsymbol{q}}(s) = 1.5\boldsymbol{q}_0(s)$. $\boldsymbol{p}_i(s)$ is parametrized as a polynomial B-spline of order 2 with 5 internal knots. We use the ellipsoidal tube approximation and consider three cases listed in Table I. The first case allows only a deviation from the path in the $r_{\boldsymbol{n_1}}$ direction and almost none in the $r_{\boldsymbol{n_2}}$ while $r_{\boldsymbol{n_1}}$ is parallel to the horizontal plane (defined by $\boldsymbol{v}$). Hence, in approximation, only a deviation in the horizontal 2D plane is allowed. The second case allows a deviation in the $r_{\boldsymbol{n_2}}$ direction and almost none in the $r_{\boldsymbol{n_1}}$ direction, while $r_{\boldsymbol{n_2}}$ makes a 45° angle with the horizontal plane. The third case allows equal deviation in all directions. Figure 8 shows the central path $\boldsymbol{x}_0(s)$ (dashed line), the optimized Cartesian path $\boldsymbol{x}(s)$ (thick full line) and the tube boundaries (thin lines). Table I lists the optimal tube following motion time T^*_{tf} for the three cases, while the motion time of the path following problem is equal to $T^*_{pf} = 0.1703$s. All three cases have lower motion time compared to the path following problem while case three has lower motion time compare to case one and two, due to the bigger freedom of the Cartesian path.

V. Discussion

We have transformed the path following problem into a tube following problem. This allows for faster motions by deviating from the given path within the allowable tolerance. By using the same transformations as in the path following problem, and by deriving efficient tube constraints, we have shown, by means of numerical simulations, that the tube following problem is still very efficient to solve. Further research will deal with exploiting robot redundancy into the optimization. This can be done by allowing such an amount of freedom on the joint paths $\boldsymbol{q}(s)$, such that the optimization exploits the redundancy to determine the fastest path inside the tube.

[1]Phantom is a trademark of SensAble Technologies, Inc. of Cambridge, MA.

	$(r_{\boldsymbol{t}}, r_{\boldsymbol{n_1}}, r_{\boldsymbol{n_2}})$	$\boldsymbol{v}$ (Sec. III-B)	T^*_{tf}
case 1	$(0.05, 0.005, 10^{-6})$	$(0, 0, 1)^T$	0.1698
case 2	$(0.05, 10^{-6}, 0.005)$	$(0, \sqrt{1/2}, \sqrt{1/2})^T$	0.1693
case 3	$(0.05, 0.005, 0.005)$	$(0, 0, 1)^T$	0.1658

TABLE I
THREE TUBE FOLLOWING CASES WITH ELLIPSOIDAL TUBE APPROXIMATION, WHERE $T^*_{pf} = 0.1703$.

References

[1] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[2] K. Shin and N. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *Automatic Control, IEEE Transactions on*, vol. 30, no. 6, pp. 531–541, 1985.

[3] W. Van Loock, S. Bellens, G. Pipeleers, J. De Schutter, and J. Swevers, "Time-optimal parking and flying: Solving path following problems efficiently," in *IEEE International Conference on Mechatronics, Vicenza, 27-28 February, 1 March*, 2013.

[4] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, 1992.

[5] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast Direct Multiple Shooting Algorithms for Optimal Robot Control," in *Fast Motions in Biomechanics and Robotics*, (Heidelberg, Allemagne), 2005.

[6] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2318 –2327, oct. 2009.

[7] D. Verscheure, M. Diehl, J. De Schutter, and J. Swevers, "On-line time-optimal path tracking for robots," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, (Piscataway, NJ, USA), pp. 610–616, IEEE Press, 2009.

[8] F. Debrouwere, W. Van Loock, G. Pipeleers, M. Diehl, J. Swevers, and J. De Schutter, "Time-optimal robot path following with cartesian acceleration constraints: a convex optimization approach," in *Mechatronics Forum, LINZ, Austria*, pp. 469–475, 2012.

[9] M. W. Spong, *Robot Dynamics and Control*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1989.

[10] W. Van Loock, G. Pipeleers, and J. Swevers, "Time-optimal quadrotor flight," in *European Control Conference, Zurich*, pp. 1788–1792, 2013.

[11] J. Andersson, J. Åkesson, and M. Diehl, "CasADi – A symbolic package for automatic differentiation and optimal control," in *Recent Advances in Algorithmic Differentiation* (S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, eds.), Lecture Notes in Computational Science and Engineering, (Berlin), Springer, 2012.

[12] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[13] M. C. Cavusoglu and D. Feygin, "Kinematics and dynamics of phantom model 1.5, haptic interface," Tech. Rep. UCB/ERL M01/15, EECS Department, University of California, Berkeley, 2001.